CERTIFICATION EXPERIENCE OF THE JAGUAR FLY-BY-WIRE DEMONSTRATOR AIRCRAFT INTEGRATED FLIGHT CONTROL SYSTEM

by

K. S. Snelling
Engineering Manager
Combat Aircraft Controls Division
Marconi Avionics Limited
Airport Works
Rochester
Kent
England

SUMMARY

This paper briefly describes the digital Integrated Flight Control System (IFCS) developed for the Jaguar Fly-By-Wire (FBW) demonstrator programme, identifying the specification requirements, resultant architecture, implementation and the incorporated self test capability. The redundancy management aspects of the IFCS are described together with the techniques for providing the pilot with relevant information to determine the IFCS redundancy status. Particular emphasis is given to the software definition and preparation procedures, and the comprehensive integrity appraisal leading to flight clearance of the system.

Following the extensive rig proving of the system, the early phases of flight test were very successfully carried out using the fixed gain control laws. During this period a major software update was commenced to incorporate the scheduled gain control laws and to enhance the self test capability. The software segregation introduced at this stage is described, together with the experience obtained in recertifying the system. Flight testing of the scheduled control laws is continuing, and the minor problems encountered are mentioned. A further software revision to include the control laws for the statically unstable aircraft is well advanced, and the benefits of software segregation identified during this revision are described.

The reliability of the aircraft and IFCS have proved, to date, to be excellent. Thus practical in-flight results of the systems ability to absorb and survive fault conditions are minimal. The redundancy management and integrity experience provided by the programme has therefore principally been in the theoretical analysis supported by controlled experimentation on the rig. These exercises have highlighted key areas of the system and software design techniques which enable these aspects to be fully and economically evaluated. These areas are described, with mention of how these techniques are being developed to simplify and improve the exercise for future high integrity digital flight control systems.

1. AIRCRAFT AND SYSTEMS DESCRIPTION

1.1. Aircraft

The FBW Jaguar demonstrator aircraft is a modified single seat SEPECAT Jaguar. Internal modifications were made to accommodate the IFCS computers and extensive instrumentation, and all of the original mechanical control rods, autostabiliser equipment and powered flight control units were removed. A third Transformer Rectifier Unit (TRU) was added to cover the additional loading of the fly-by-wire system and instrumentation, and revised power distribution was introduced to meet the power supply integrity requirements of the IFCS.

Three independent 28V bus bars, each battery backed, are supplied by the three TRU, and each computer of the IFCS consolidates power from two of these bus bars as shown in Figure 1.

The two engine driven hydraulic pumps were replaced by units with greater capacity, and the emergency electrohydraulic pump was replaced by two pumps of greater capacity each driven by one of the independent, battery backed, 28V bus bars. These provide two independent hydraulic systems, each with an emergency supply primarily to power the flying control actuators, the system including provision for priority valves if found necessary. The standard power transfer unit, allowing transfer of power but not fluid between systems, is retained.

Externally the aircraft is little changed, though later in the trials programme leading edge strakes will be fitted along the air intake boxes. Provision is made for fixed ballast to be carried in the rear fuselage, and this together with fuel management procedures enable the centre of gravity to be moved aft to give a manoeuvre point of $-3\%\bar{c}$ to $-5\%\bar{c}$. The leading edge strakes will move the centre of pressure forward to give a manoeuvre point of $-10\%\bar{c}$.

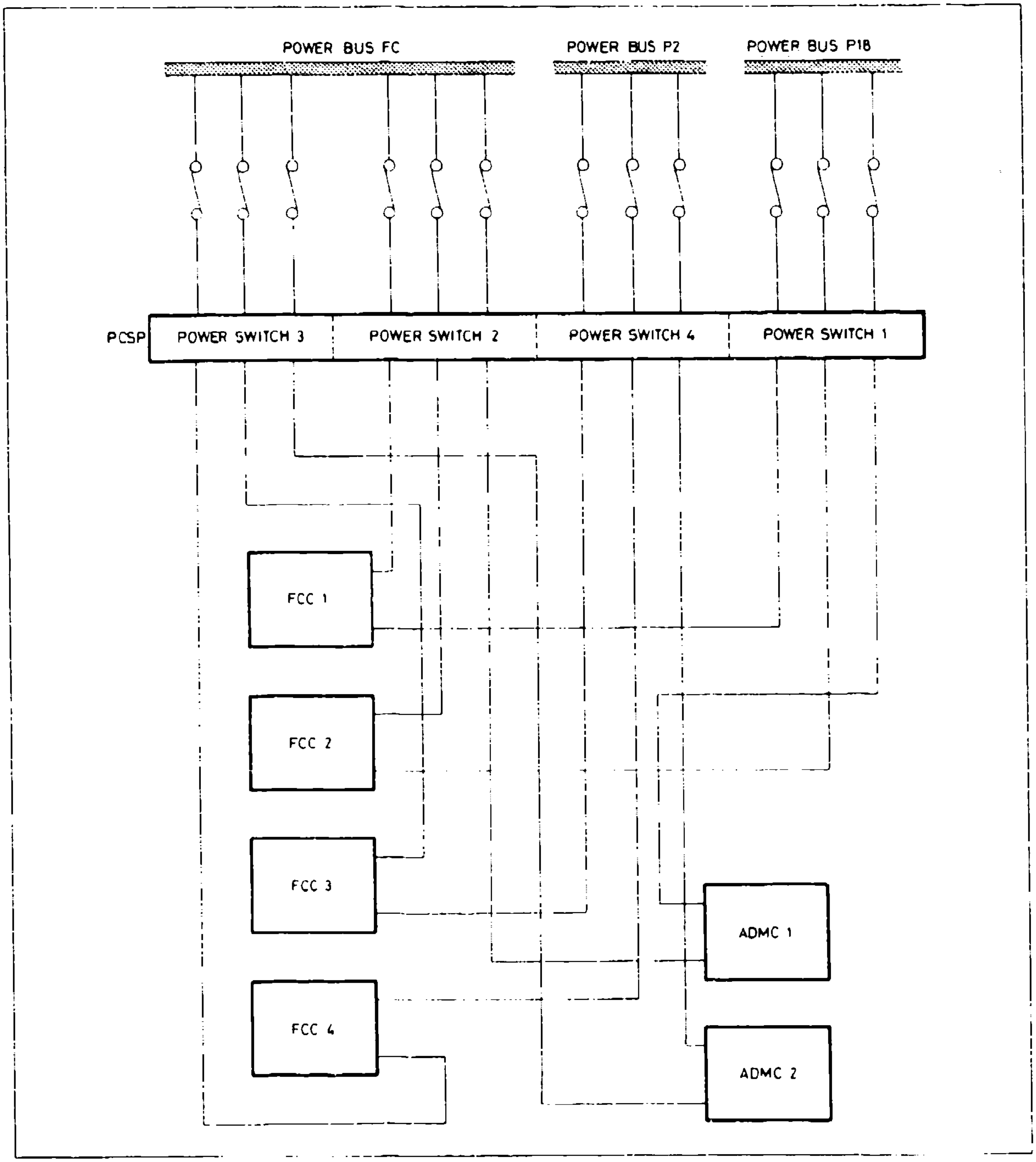

Figure 1 Flight Control System Primary Power Distribution

1.2. Integrated Flight Control System

The system architecture shown in Figure 2 was evolved to meet the following specification requirements.

- Overall system loss probability (including first stage actuation) shall be no greater than 10^{-7} per hour.
- Any two electrical failures in the system shall be survived.
- The electrohydraulic first stage actuation would have only two independent hydraulic supplies with no interconnection between them.
- The system shall survive a hydraulic system failure followed by an electrical system failure or an electrical failure followed by a hydraulic failure.
- The system shall rely on majority voting of the redundant elements for failure survival rather than self-monitoring within each redundant element.
- Similar redundant digital implementation shall be adopted without any reliance on any back-up flight controls (e.g. mechanical or analogue links).

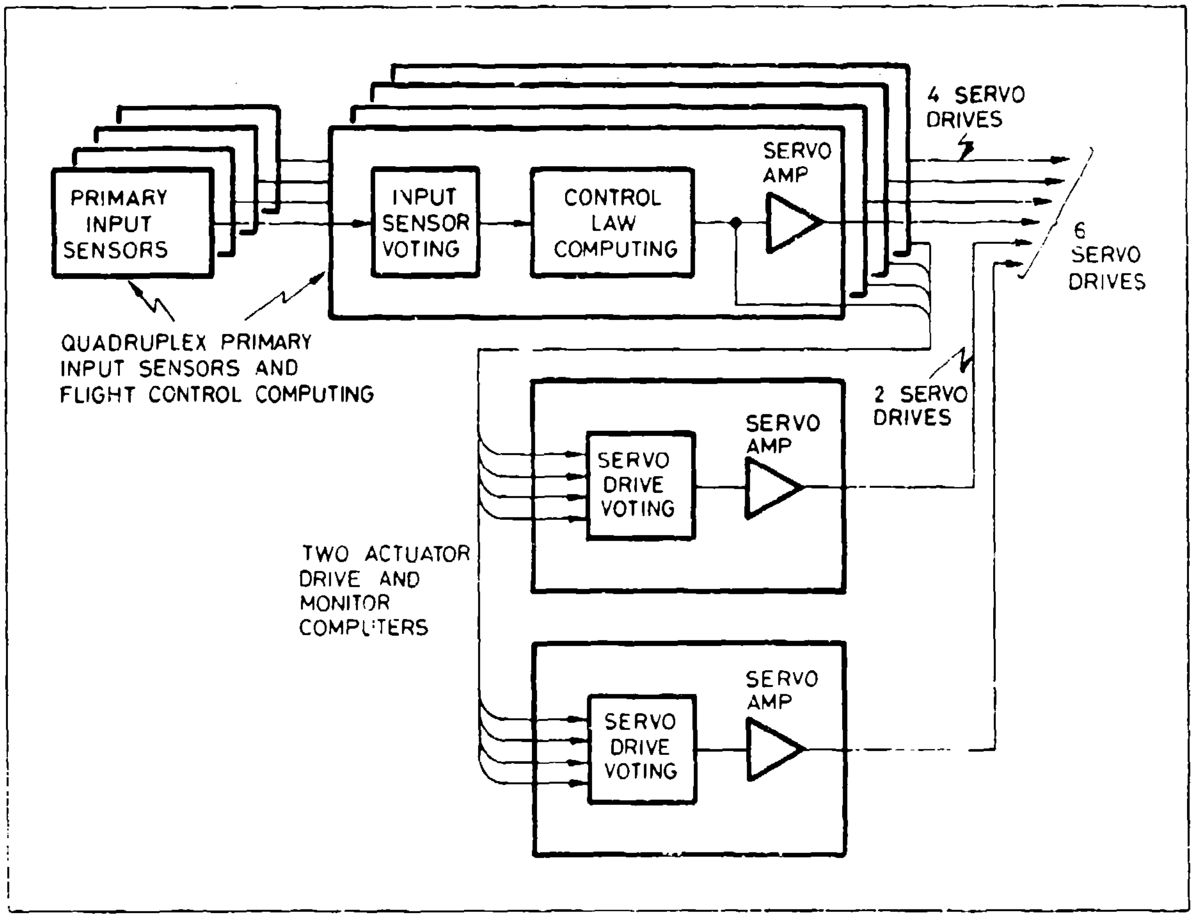

Figure 2 System Architecture

These requirements led to the incorporation of the duo-triplex actuation system, developed by Dowty Boulton Paul, to drive the rudder, two taileron and two spoiler control surfaces. The five Powered Flying Control Units (PFCU) are essentially similar, with variations in valve ports, jack strokes and diameters. Each PFCU, schematically shown in Figure 3, contains six flapper nozzle servo valves which convert electrical inputs from the Flight Control Computers (FCC) and Actuator Drive and Monitor Computers (ADMC) into hydraulic signals which are then used to drive a pair of first stage spool/control valves. Each servo valve is connected to two pairs of opposing pistons inside one of these first stage actuators. The pistons act on flanges mounted on the actuator spools, two pairs being used to prevent assymmetric loading. Both flanges are therefore driven by six pairs of opposing pistons, two pairs from each of three servo valves. A mechanical link between the two actuators ensures that the spools and thus the flanges move in unison, so that all six servo valve outputs are effectively summed together. In this way failures in at least two lanes can be absorbed within the actuators, the four good lanes overriding the failed lanes.

A separate hydraulic supply feeds each trio of servo valves associated with the first stage actuators and is also routed through the first stage actuator to the corresponding jack of the conventional tandem power control unit. Thus failure of either hydraulic supply can be tolerated by the PFCUs in addition to at least one electrical failure that affects the computing driving the side of the actuator unaffected by the hydraulic fault.

This actuation architecture requires 6 independent drive signals to each actuator, but the remaining integrity objectives do not necessitate the cost and complexity of a full six lane system. The Flight Control System (FCS) is therefore essentially a quadruplex digital system with special facilities to provide the additional independent drives to the actuators. All mechanical rods downstream of the trim and feel units have been removed, thus there is no mechanical or independent back up reversion.

Quadruplex Position Sensors (QPS) are used to sense pilot control demands in terms of stick, pedal and trim inputs and quadruplex rate gyros sense aircraft pitch, roll and yaw rates. Four identical digital FCC are used to process these signals together with those from other sensors. The resulting command signals are used to control the actuators. To convert the quadruplex signals from the FCCs into the sextuplex signals required by the actuators, the FCCs are supplemented by two ADMC.

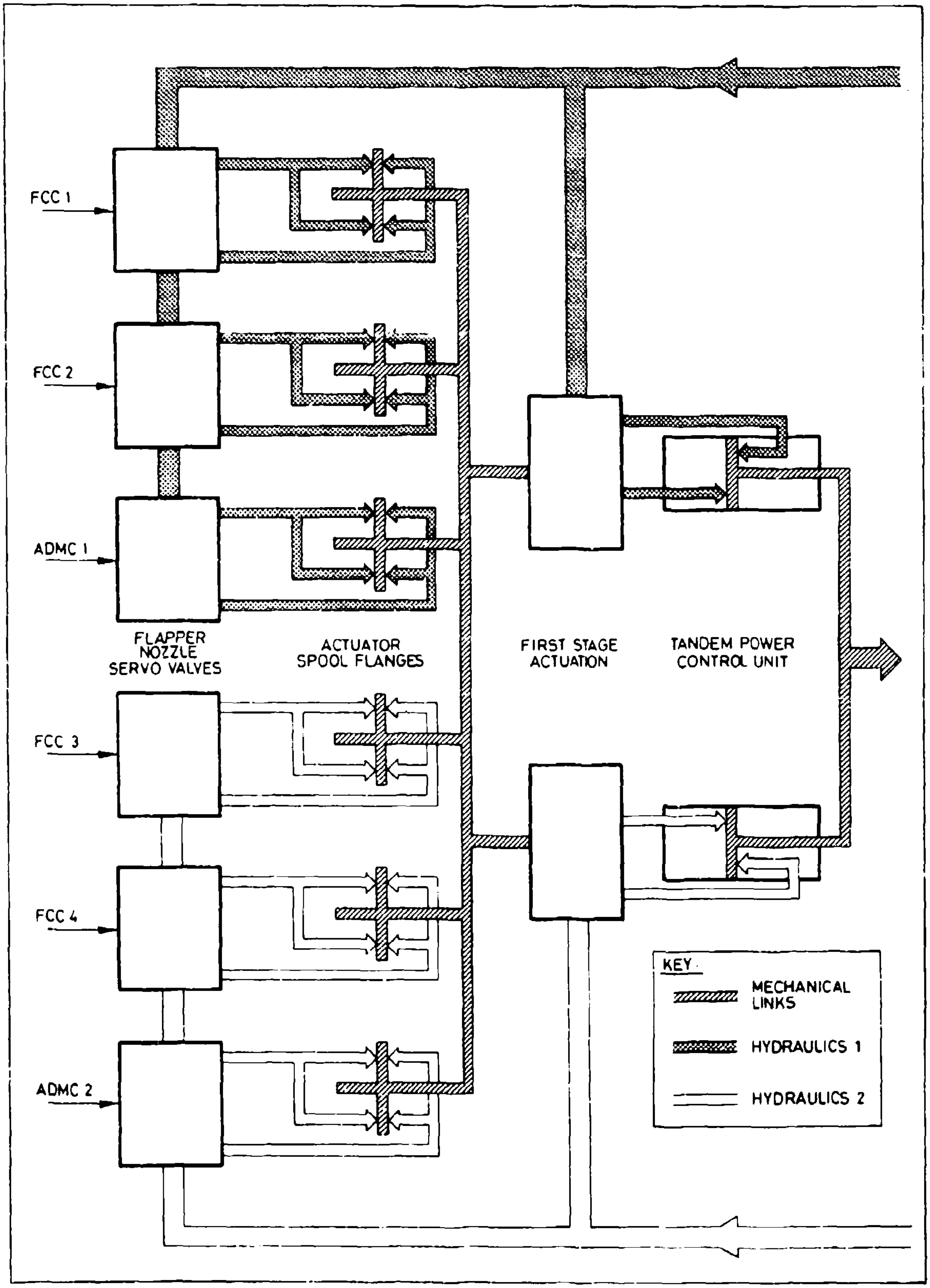

Figure 3 Duo-triplex Actuator Scheme

Figure 4 shows the schematic of an ADMC which receives optically coupled signals from all four FCC, converts them to analogue and votes them to provide a consolidated, essentially independent output.

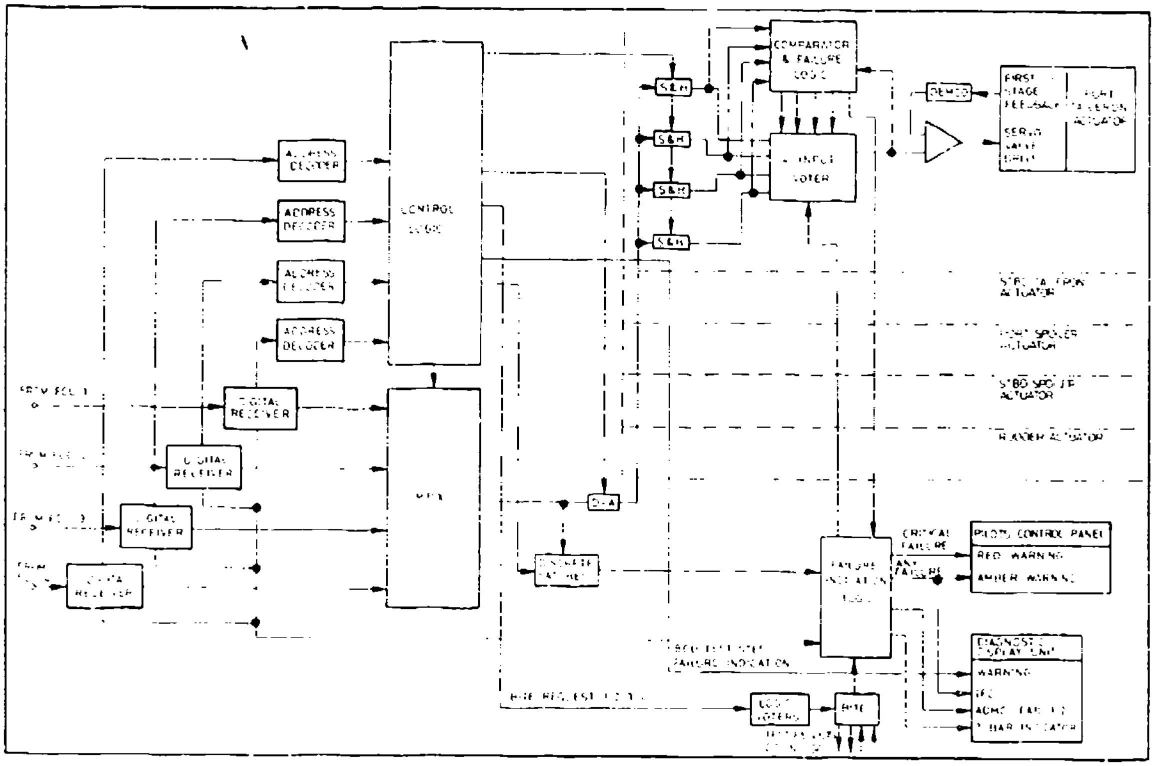

Figure 4 Actuator Drive and Monitor Computer Schematic

The Jaguar FBW system configuration is illustrated in Figure 2 which presents a simplified schematic of the primary control path. In addition to the quadruplex primary input sensors, sensors of lower redundancy are used for those functions which may be necessary for optimum handling qualities but which are not necessary for safe flight. These are dynamic pressure, static pressure, incidence and sideslip which are all triplex sensors; and lateral acceleration, flap position and airbrake position which are duplex sensors. Triplex dynamic and static pressures are provided by three pitot static probes (the standard nose boom and two side mounted probes). Triplex incidence and sideslip signals are provided by four Airstream Direction Detector probes (ADD) mounted around the nose of the aircraft.

The FCS also uses a number of quadruplex and duplex discrete inputs for switching functions. A simplified overall system configuration is illustrated in Figure 5. Cross lane data transmission is achieved via dedicated, optically coupled serial data links as shown in Figure 6. Voting and failure rejection logic in each computer maximises the system failure absorption capability and ensures the the system is able to survive two sequential failures of all primary input and feedback sensors. The system is designed to run synchronously, but has been operated assynchronously for considerable periods without observable degradation of performance. A more detailed description of the system architecture and the system LRUs can be found in reference 1.

The system includes comprehensive Built-In-Test (BIT) features which were specified to provide an accurate, decisive, and repeatable method of measuring equipment functional characteristics. In particular the BIT is used to clear the system in the aircraft prior to each flight thus its integrity and fault detection ability have to be compatible with the overall integrity of the system. The facility developed has met the objectives and provides an invaluable aid to FCS commissioning on the aircraft and reclearance of the FCS following Line Replaceable Unit (LRU) changes. A pilots Control and Switch Panel, shown in Figure 8, provides system status indication to the pilot. Status signals from the Flight Control Computers are consolidated to illuminate a STATUS amber warning (first failure) or red warning (similar second failure). The pilot may attempt a reset, when an amber warning is given, by pressing the STATUS button. If the detected disparity is no longer present the system will return to full operation status and the warning is extinguished. A red warning inhibits the status reset facility. Separate status indicators are provided for the secondary sensors. The panel also carries the autopilot engage buttons, the BIT initiate button, a facility to select different control law gains, and power switches to isolate the supplies to the computers to enable pre-flight check of the power supply consolidation within these units.

The FCS Equipment has been developed to production standards, as shown in Figures 7, 8, 9 and 10, and qualification tests have been completed.

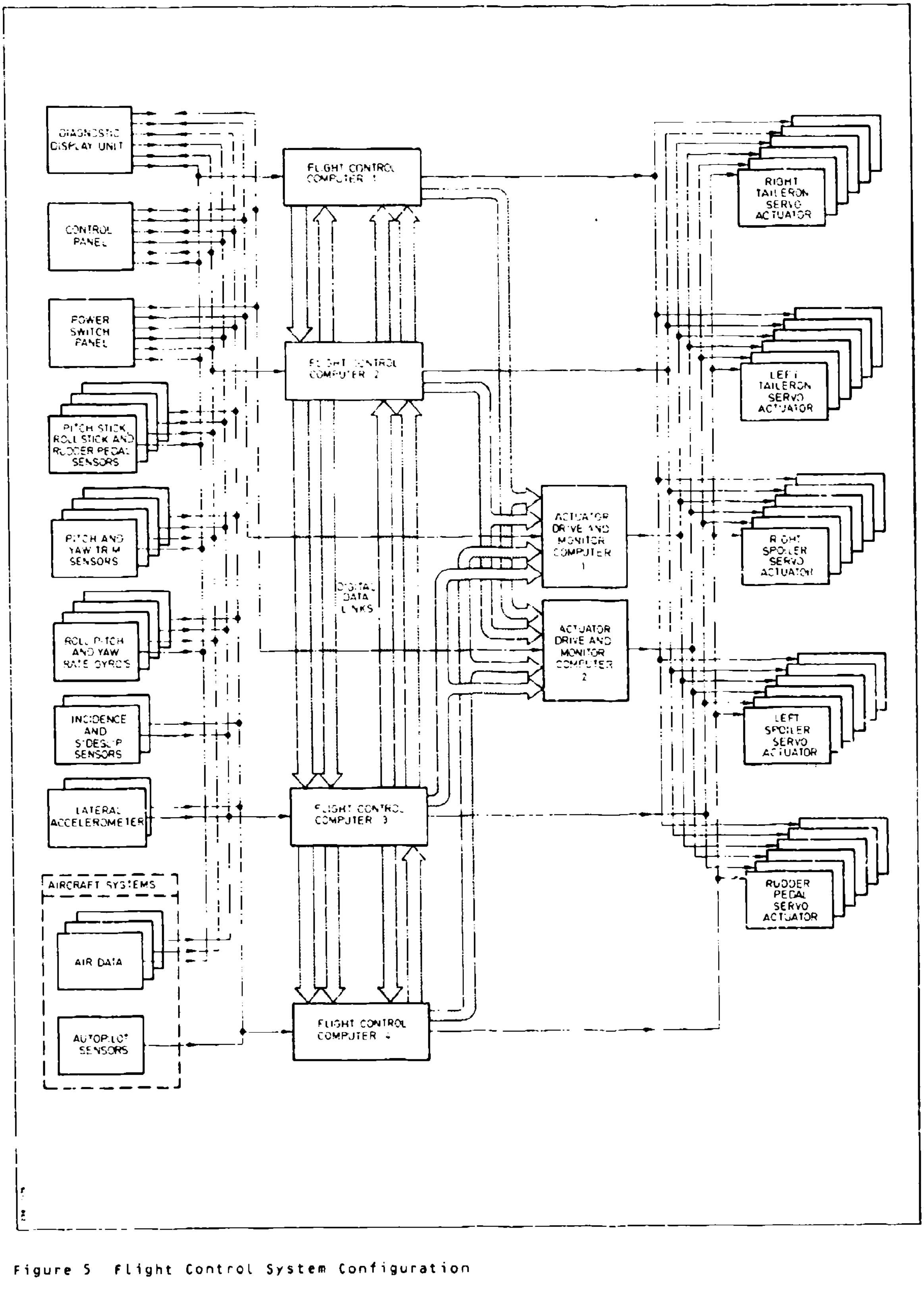

Figure 5 Flight Control System Configuration

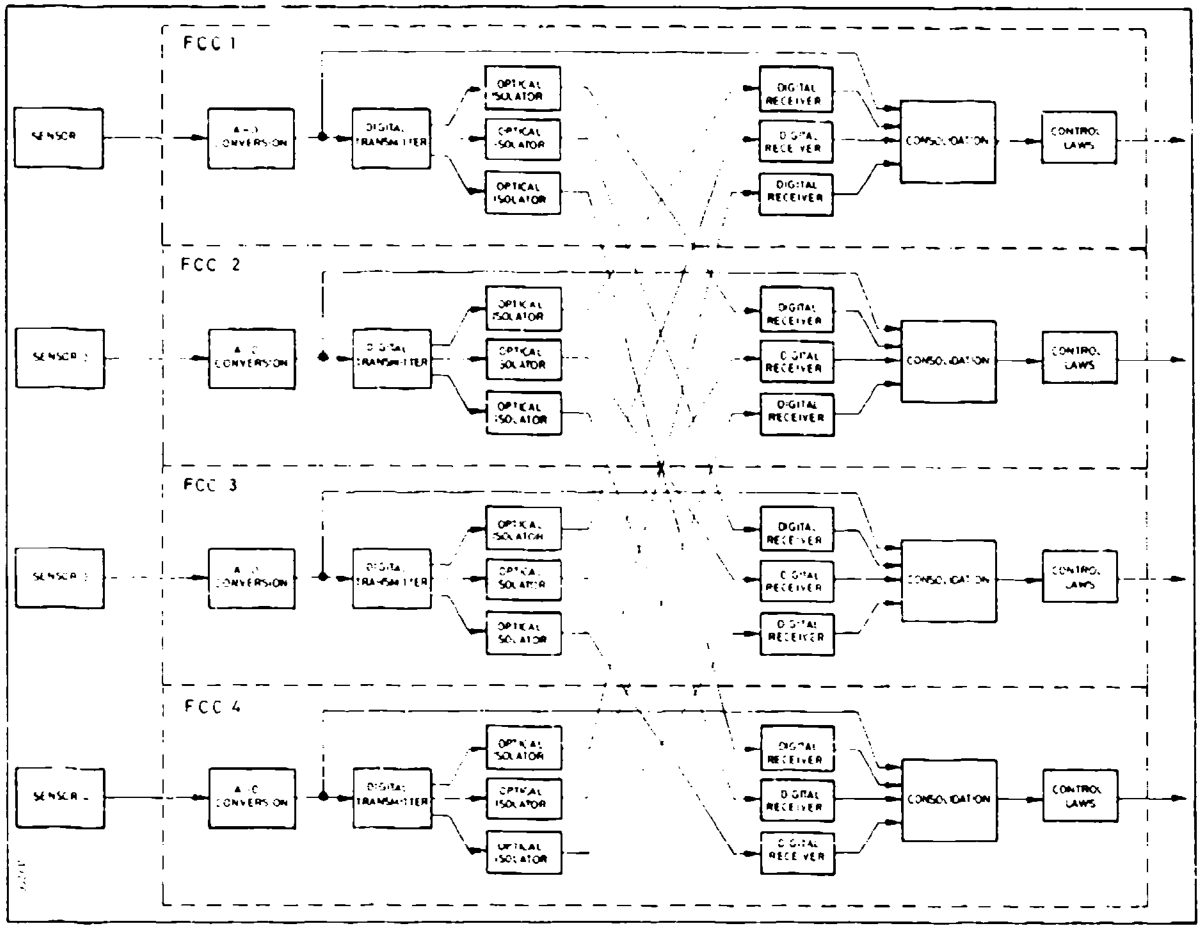

Figure 6 Optically Coupled Cross Lane Data Transmission

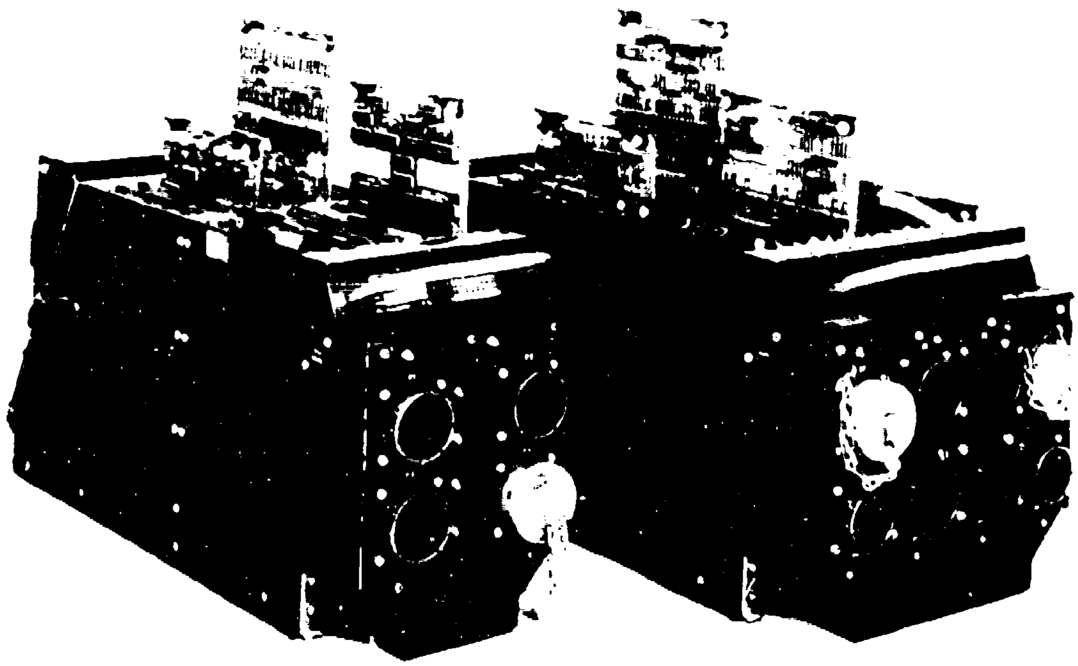

Figure 7 Actuator Drive and Monitor, and Flight Control Computers

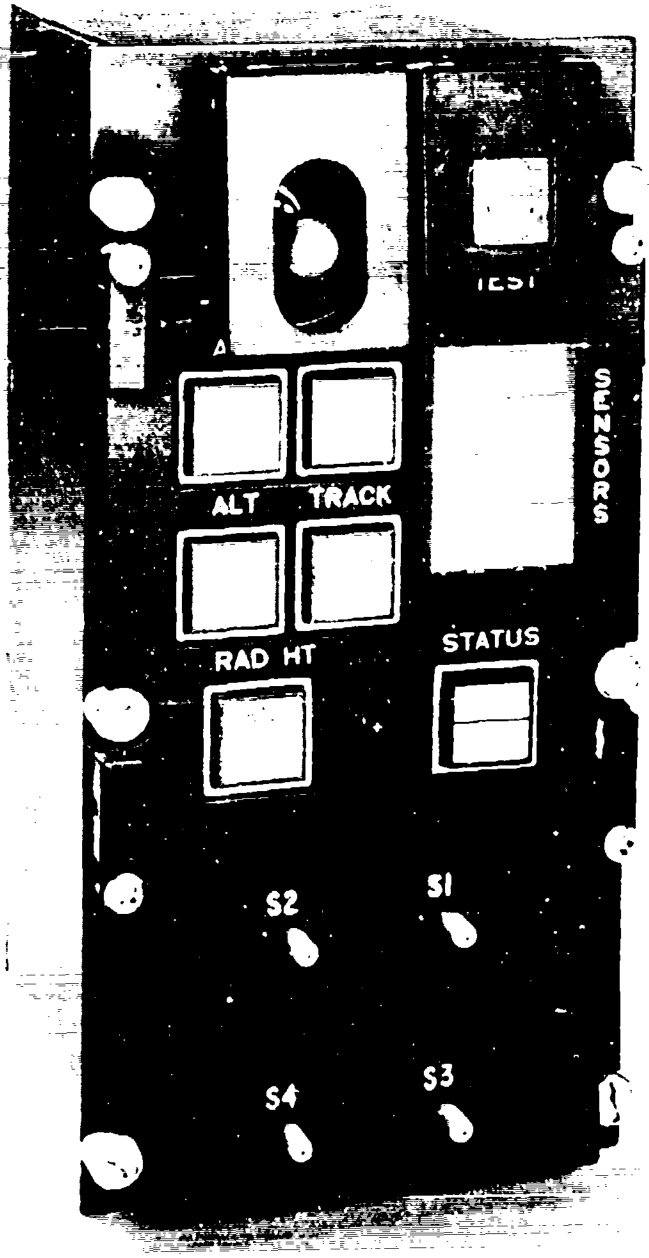

Figure 8 Pilots Control and Switch Panel

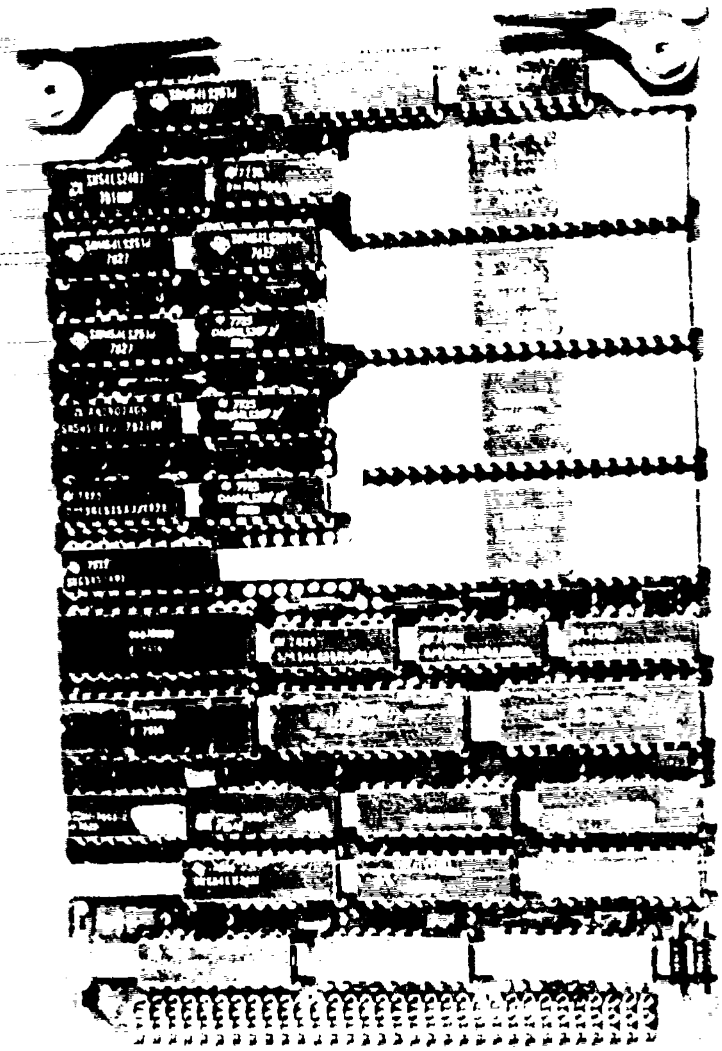

Figure 9 Typical Computing Module

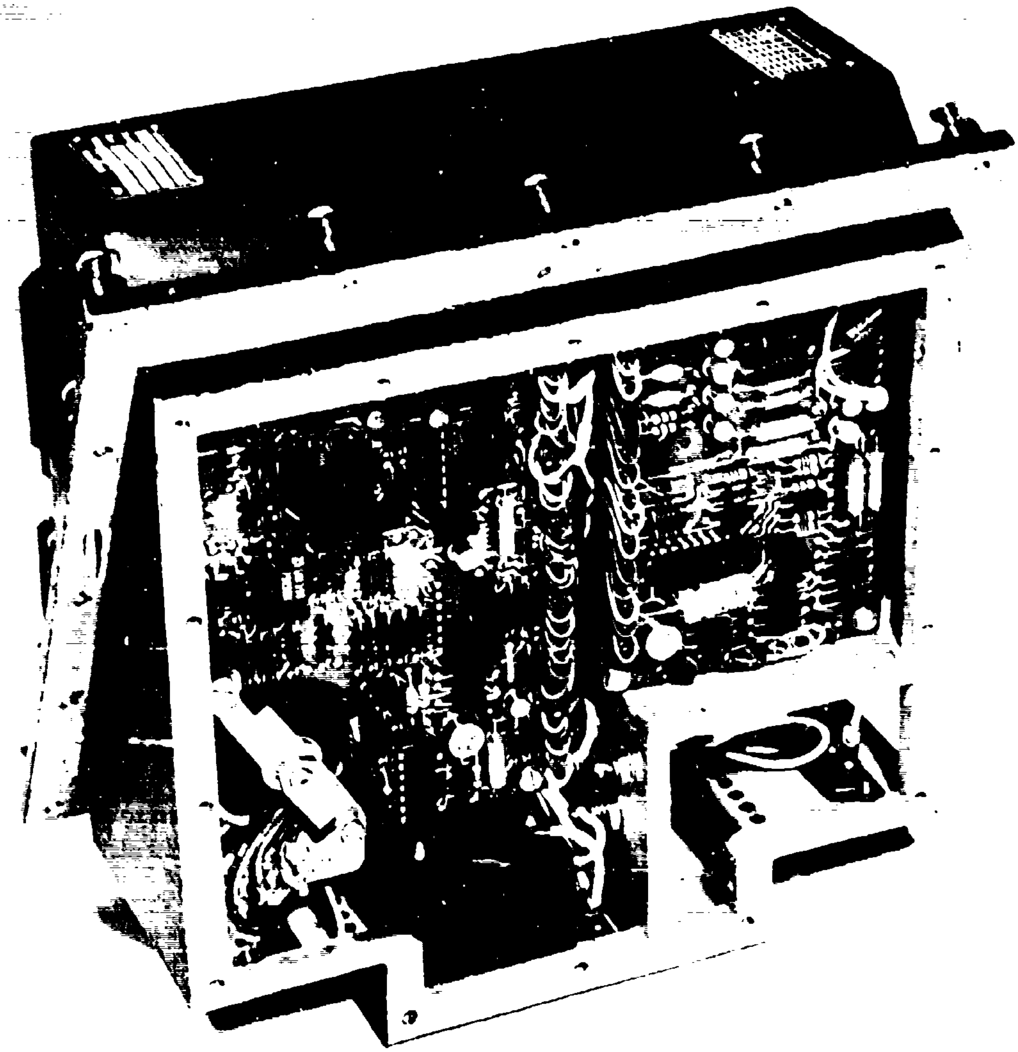

Figure 10 Power Supply Unit

2. FLIGHT RESIDENT SOFTWARE

2.1. Introduction

Both the system specification requirements, and cost/timescale considerations dictated the use of common high integrity software in all lanes of the FCS. There is, therefore the possibility of introducing design limitations via the software that could result in a common mode malfunction of the system and a subsequent safety critical loss of control. To contain this problem software structures and design procedures have been evolved over several digital FCS programmes. These maximise the visibility of the software to facilitate thorough test and functional audit during the design phase. These are supplemented by clear requirements definitions, detailed documentation, and rigorous production and configuration control procedures.

2.2. Flight Software Organisation

The real time control is achieved by a hardware Master Reset Timer which calls a non interruptable Executive. The Executive then calls the Frames (processing time slices containing related functional modules) in a defined sequence to provide the required iteration rates for the various computing paths. Each Frame typically contains control laws, with related signal selection and logic module functions, and consists of a set of program modules each defining a function that is easily defined, implemented, tested and audited. The worst case run time of a Frame is controlled at the design stage to ensure that the computing task is completed before the Master Reset occurs. Should any fault occur that causes the Frame run time to exceed the Master Reset time interval, this is detected and flagged as a computer fault.

The structure of the flight resident program is shown schematically in Figure 11.

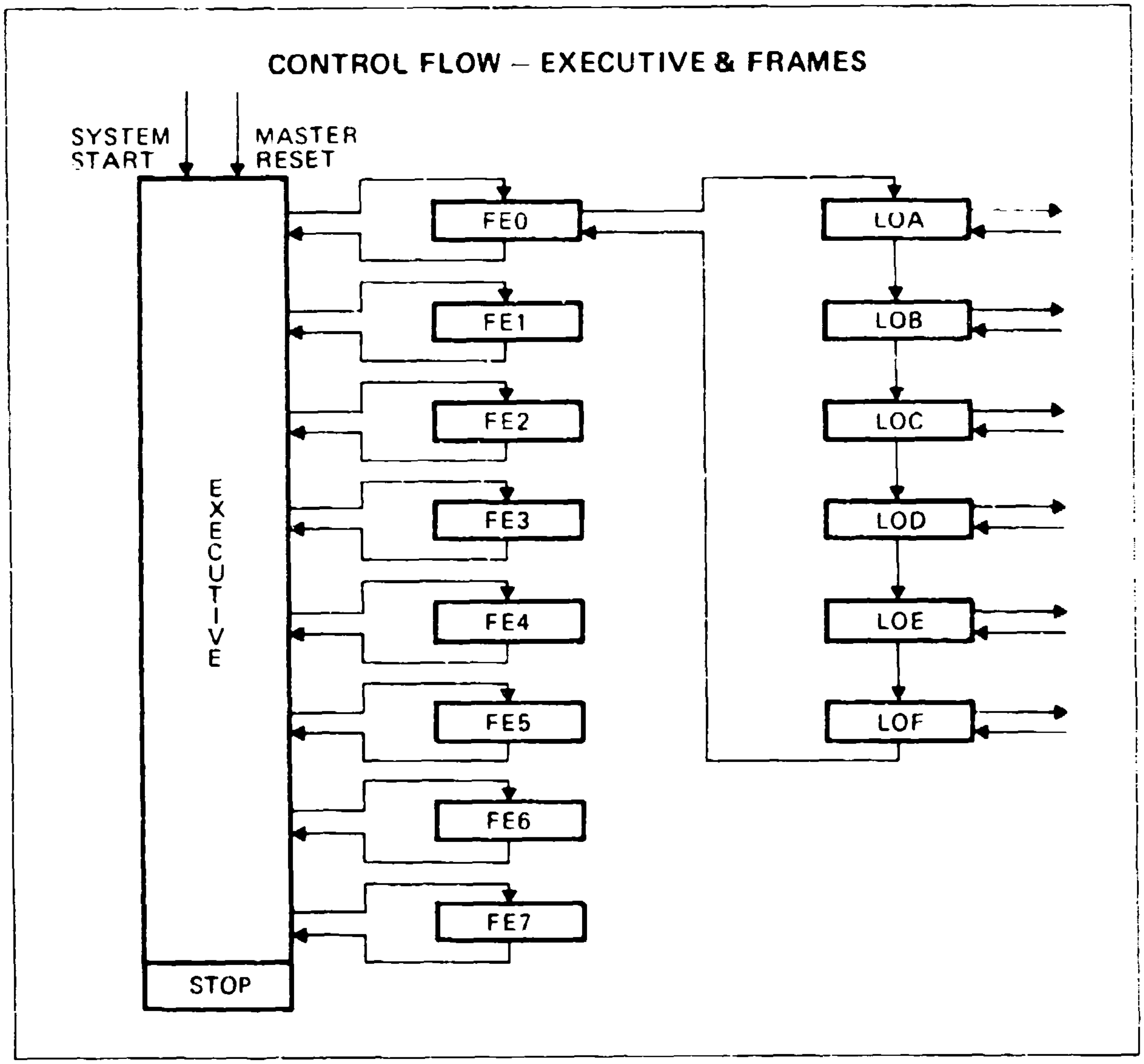

Figure 11 Flight Resident Program Structure

2.3. Flight Software Development Process

The key documents controlling the software design are the Software Requirements Document (SRD) and the Software Structure Document (SSD), prepared in conjunction with BAe from their basic software specification, control law definition and interface documents.

The SRD uses English language and program statements to define the design implementation. These statements are formed to eliminate definition ambiguity and form the basis of definitive software design specifications which are testable to prove the accuracy of the definition.

The SSD defines the running order of the modules within the program segments. The structure is designed to ensure that chronological flow of data from input, through processing, to output is in strict sequence.

An important aspect of the initial software design process is the definition, optimisation and validation of the frequently used algorithms, particularly those associated with system integrity such as signal consolidation and monitoring. MAv developed 6 different voter monitor algorithms to cover the range of analogue and discrete signals at various redundancy levels, together with many other filter and schedule routines.

The codes of practise used in designing and testing the Flight Resident Software FRS are defined in the Programmers Manual and the Testers Manual. These also define the procedures and documentation requirements for configuration and quality assurance control.

The target processor structure, input/output requirements, and the task orientated instruction set, are also rigorously defined.

The overall software development process is shown diagrammatically in Figure 12. The software requirements documents are interpreted to produce software module Design Specifications, which include definition of the module implementation in the form of FORTRAN statements. These high level language statements are then coded into the macro assembler statements used by the FCC processor, supported by FORTRAN comment statements to improve code visibility. A library of well proven macros has been established which

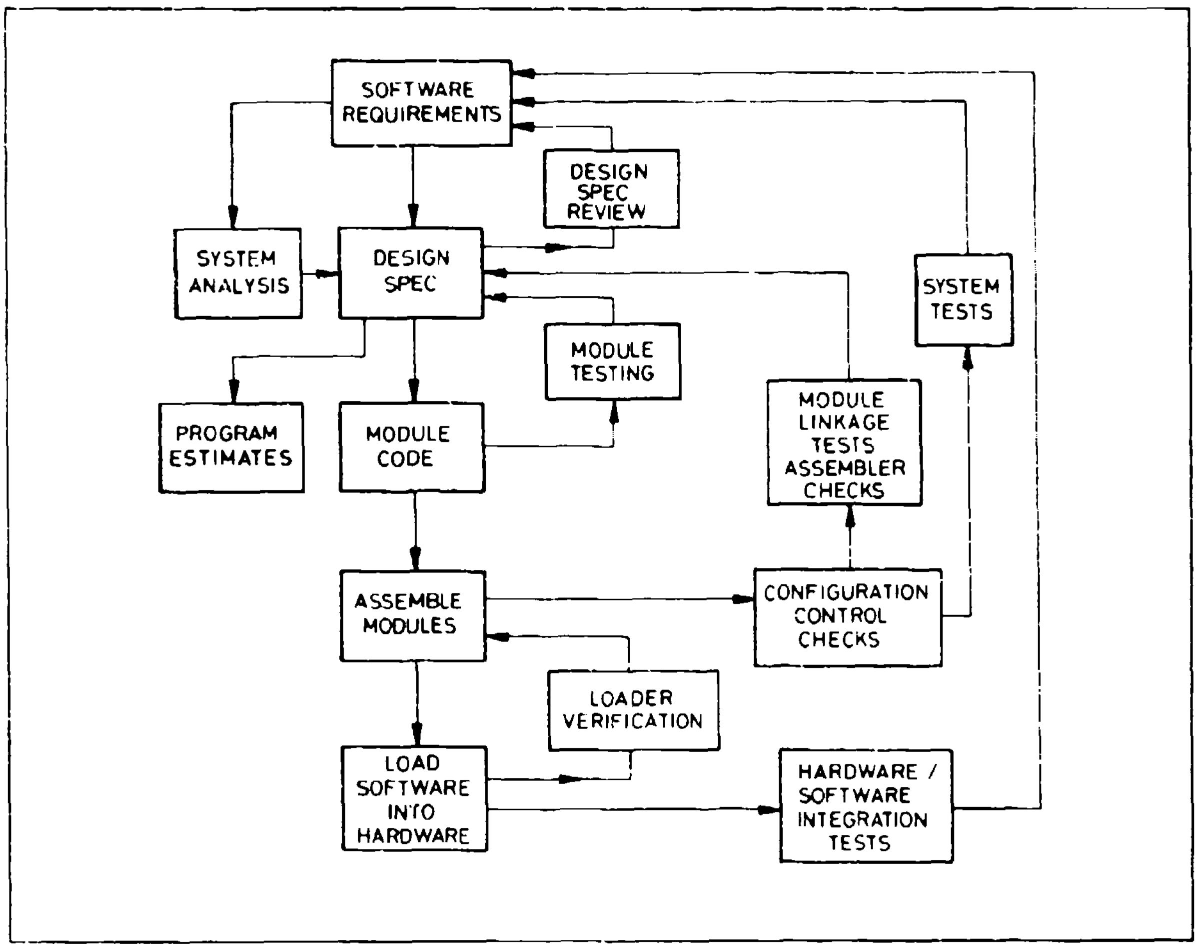

Figure 12 Software Development Process

covers some 70% of the data management and control law software requirements. A corresponding Design Report is produced listing the assembled code for the module together with details including run time, storage requirements, design programmer, module ident, progress card reference, and relevant design calculations. A Module Test Specification is written by another programmer who has neither designed nor coded the relevant module. This procedure minimises the possibility of carrying a module design error through to the test specification. The module code is then tested on a host computer using the specified test harness, and the results are recorded in a Module Test Report.

The module documentation is audited by senior programmers to ensure that the code accurately represents the design requirements and that all design rules such as single entry, single exit, all decision logic in the forward direction etc. have been observed. The audit also ensures that the test rules have been followed including all paths through the module have been exercised and that sufficient intelligent testing has been defined to check overflow/saturation conditions for the module. The test results are correlated with the test specification requirements to ensure all tests are complete and accurate, and the documentation is checked for completeness.

When all the modules are completed the code is assembled into the frames and then the full Flight Resident Software (FRS) with similar testing, reporting and audit at each stage. End to end tests are carried out on the fully assembled programme using the host computer before generation of the PROM device code for transferring the FRS to the target computers. At this stage the Quality Assurance department complete their audit of the software preparation process, check the PROM device code review the design and configuration documentation, and if all is satisfactory release the software for formal issue.

The development and testing of high integrity FRS for Flight Control Systems has been carried out on several host computers using 'in house' developed software tools progressively enhanced, and proven by duplicate assemblies on successive host computers. The result is a suite of well proven support programmes. These programmes include the macro expander, assembler, simulator, PROM code generator and test result annotator. Each includes routines to check valid usage of instructions, storage, work space, run time etc. Any deviation from the rules in these areas inhibits the generation of the final code and the PROM device code.

The SRD, SSD, design and test documents, Programmers and Testers Guides, the generated code and the host computer software are under strict configuration control from the initial issue. Changes can only be introduced by formal Change Requests which are authorised by the Chief Programmer and the Project Manager. Build Standards identify the documentation issues and Change Requests applicable to each issue of the software.

The production of the software is controlled using Progress cards which are identical to those used for controlling manufacture of hardware. These cards create a historical record of all stages of the software development, and the identities of the programmers completing each task. All relevant Change Requests are recorded on the card which can be used to trace the development of the module through all design, test and analysis phases.

Strict adherence to the above techniques generates highly visible FRS, fully audited, well tested and inherently of the required integrity.

3. INTEGRITY APPRAISAL

The complexity, novelty and specified requirements for the IFCS necessitated a major work programme to appraise the resultant integrity. The technique employed analysed the system integrity assuming perfect implementation, and subsequently audited the implementation to assess the effects of possible faults and design defects.

The integrity of the IFCS is primarily determined by the system architecture. Therefore the elements of maximum concern are the points where the redundant lanes are consolidated or otherwise connected, together with the potential for common mode safety critical design defects in the hardware, firmware or software.

The appraisal was carried out using both 'bottom up' and 'top down' analyses, and since some of the issues involved could not lead to useable quantitative estimates of risk, qualitative assessments were also necessary.

The main elements and interactions of the appraisal/audit methodology are shown in Figure 13 and included:-

i) 100% coverage single fault Failure Modes and Effects Analysis (FMEA).
ii) Multiple fault FMEA for specific combinations.
iii) Flight resident software audit.
iv) Appraisal of special areas.
v) Configuration inspection.
vi) Qualification programme.
vii) Burn-in programme.

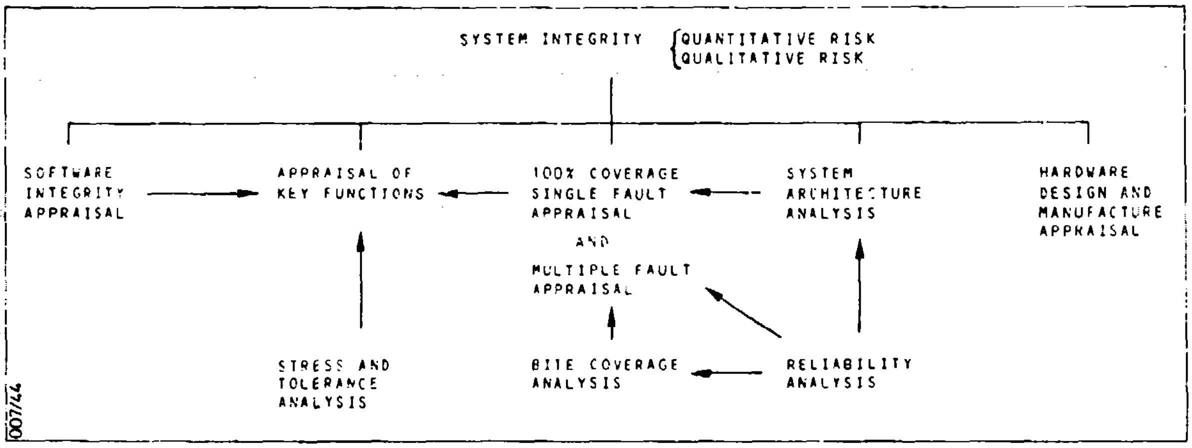

Figure 13 Integrity Appraisal

These primary elements were supported by

a) Module, chassis and LRU FMEA.
b) Microprogram appraisals.
c) Voter/monitor appraisals.
d) Tolerance analyses.
e) BITE coverage analyses.
f) System architecture analyses.
g) Reliability analyses.

During the course of the appraisal detailed technical evaluations of various features and functions of the IFCS were made. The requirements for these evaluations were generated mainly from the FMEA activity, and by BAe as a result of their test activities. These evaluations were reported as a series of Technical Appraisals appended to the overall integrity report, and their results incorporated into the risk assessment.

The integrity appraisal was conducted by a team with specialist knowledge of the equipment design, but to ensure rigour in the appraisal they reported to an independent authority consisting of senior engineers from MAv and BAe.

An essential part of the system clearance depended on the extensive emulator, rig and aircraft testing carried out at BAe, Warton. During these exercises, any unexpected observation that could not immediately be explained by the personnel involved in the test resulted in the raising of a formal query. A written response to every query, approved by both BAe and MAv, was a mandatory requirement for final Q.A. clearance of the aircraft for flight.

A fully detailed description of the system integrity appraisal can be found in Reference 2.

4. FLIGHT TESTING

Following extensive rig and aircraft ground trials, including a considerable amount of electromagnetic compatibility (EMC) and power supply transient testing, the first flight took place on 20th October 1981. Flight testing of the fixed gain control laws was completed in 13 flights, compared with the 14-22 flights budgetted. The aircraft proved easy and straightforward to fly with excellent FCS reliability. The flying rate of the aircraft was never limited by any problems within the FCS but solely by the large amounts of data to be analysed between each flight.

During this 4 month period only one FCS LRU was exchanged due to a defect. The LRU change was prompted, during routine servicing, by BIT detection of a spurious cross lane data transmission malfunction. No in-flight computing malfunctions occurred throughout these trials. A single inflight FCS failure warning occurred just prior to landing on Flight 13, caused by a delay in the quadruplex switch on the undercarriage selector. This switch is a standard Jaguar part, and the possible delay between operation of the two pairs of switch contacts could exceed the time specified in the interface documents.

The FCS detected this delay on a slow undercarriage selection and correctly diagnosed a virtually simultaneous similar double failure resulting in an FCS RED warning to the pilot. However the redundancy management logic successfully dealt with this situation and provided the correct mode selection to the control laws and an otherwise uneventful landing was achieved. After this particular flight, the in-flight BIT failure identification table (FIT) was interrogated via the system Diagnostic and Display Unit (DDU) and immediately identified the cause of the warning. Recurrence of the problem was prevented by a software change to increase the acceptable time delay between the operation of the switch contacts. Pilot confidence in the serviceability of the system

prior to each flight was enhanced by the thoroughness of the BIT function which is a pre-requisite for system engagement. For this demonstrator aircraft, the BIT requires pilot interaction which could be automated to a large extent in a production aircraft environment. However, even this BIT could be completed in about three minutes.

For further details of ground and initial flight testing of the IFCS see reference 3.

5. SOFTWARE REVISION AND FURTHER SYSTEM TESTING

5.1. Scheduled Control Laws for Stable Aircraft

Immediately following certification of the initial issue of FRS, a revision was commenced to incorporate scheduled gain control laws, to enhance the BIT function and to rectify problems encountered during the early trials which had not necessitated immediate correction. This proved to be a very extensive modification exercise resulting in changes to some 75% of the 400 modules comprising the FRS. However the timescale and cost of preparing the new issue was very much less than for the initial issue, and by building on the system integrity appraisal techniques established for the previous system standard, the certification was achieved with less than 20% of the effort required previously. The major changes in system performance required were achieved with only the single hardware modification which changed the contents of the programme store devices.

Recognising the problems of cost, timescale and integrity, associated with software modifications, it was agreed at this stage to be cost effective for additional software segregation to be introduced to the FRS. The 21K words of software required were partitioned across 26K words of store. This was organised not only to provide software segregation at module and segment level, but also to contain different sections of software within separate programme store devices. The objective was to enable future software changes to be contained to a minimum number of software modules and programme store devices. Thus bit for bit comparison of successive FRS assemblies would easily identify the change areas and enable subsequent verification and validation to be more localised than could be justified if the new assembly changed all of the programme store instruction locations.

5.2. Lightning Testing

Lightning protection measures were designed and built into the FCS and it's aircraft interfaces, and extensive EMC susceptibility, bulk current injection and transient testing carried out before the first flight. However, the effects of a lightning strike on an aircraft are unpredictable due to the complex interactive effects of the structure, equipment layout and cable runs. Thus for the early flight trials the aircraft was prohibited from flying in areas where lightning activity was likely. Subsequently a series of simulated whole aircraft lightning tests were carried out to evaluate the effectiveness of the design to protect the FCS from large electromagnetic pulses and thence to obtain a relaxation of the flight restrictions.

In conjunction with the Lightning Studies Unit from Culham (UKAEA) and RAE Farnborough, the tests were carried out by BAe Warton. The simulated lightning pulses were produced by discharging a high voltage, high di/dt generator into the aircraft at the base of the pitot probe. Conductors forming a frame around the aircraft were connected to various parts of the aircraft structure, e.g. tail cone or fin tip, to form the return path for the high current pulses and create the required electric field around the airframe. Extensive monitoring was employed with the measured results being transmitted to the screened recording room via fibre optic data links. Further details of these tests can be found in references 4 and 5.

Test pulses up to 80KV and 100KA were discharged into the aircraft configured into an effectively 'flight-ready' condition, with electrical and hydraulic systems powered and the FCS operating. These pulses represent moderate to severe lightning strikes yet there was no measurable or observable corruption or interference of the FCS function. This has generated considerable confidence in the design techniques used to provide lightning protection for the FCS on the Jaguar aircraft, but extrapolation of the results is necessary to prove the case for rescinding the flight restrictions. MAv are extending these tests by subjecting representative interface circuits to transient voltages defined by Culham as a result of the measurements taken during the whole aircraft lightning tests. These transients are essentially single pulses but with controlled rise, decay and damping chararteristics to accurately simulate the extrapolated effects of an extreme lightning strike.

The Jaguar Fly-By-Wire Demonstrator subsequently became the first aircraft to fly after being subjected to whole aircraft simulated lightning tests.

5.3. Flight Test of Scheduled Control Laws

The rig and early aircraft ground trials of the scheduled control laws detected several peculiarities and faults. Intermittent data transmission errors were detected during BIT, and an initially inexplicable incorrect FCS status was occasionally seen at the end of the pre-flight BIT. Several in-flight secondary sensor failures were also recorded.

The majority of these problems were easily identified and diagnosed by use of the BIT and interrogation of the Failure Identification Tables. These were corrected by attention to screening and changes of secondary sensors. However, after several early observations the problem which caused the incorrect post BIT status of the FCS became so infrequent that efforts to capture the history of events leading up to it were unsuccessful. Resolution of the problem prior to commencing the flight testing therefore became dependent on theoretical analysis of the software to predict the possible causes. The structured form of the FRS, and the achieved visibility of the code, enabled the investigation team to establish that there was only one possible way for this situation to develop. Subsequent review of the recorded facts on the incidents, and controlled tests, demonstrated beyond reasonable doubt that this analysis was correct. The situation was caused by occasionally adopting an incorrect procedure that could only be initiated when particular test equipment was connected to the system, and therefore could not occur in flight.

The objective of this phase of flight testing was to assess the aircraft handling with scheduled control laws, check training and spin recovery modes and complete the flutter envelope expansion with a modified standard of tailplane actuator. Testing of the aircraft continued with stores to reduce the manoeuvre margin in preparation for subsequent relaxed stability and unstable flight trials. At the time of writing this paper these trials were approaching a successful conclusion.

5.4. Scheduled Control Laws for Unstable Aircraft

Further revision of the software was required to incorporate the control laws to optimise aircraft performance in the unstable configuration created by addition of ballast and fuel management techniques. This revision required much less change than the previous revision, therefore overall comparison of the tasks cannot be used to assess the benefits obtainable from the introduction of segregation. However at the individual change level, clear benefits have been observed. This is particularly the case for late changes or corrections which could be isolated to a single programme store device change.

Significant reduction in FRS modification time, PROM code generation and hardware reprogramming has been achieved. Combined with increased confidence in the fidelity of the unchanged parts of the programme, these have dramatically reduced the time to introduce and prove late changes immediately prior to the formal validation and verification process. As yet the programme has not reached a stage where formal recertification of the system after a small FRS revision has been attempted. It is not, therefore, possible to state the benefits that segregation provides for this activity, but it is predicted that these could be very significant.

Flight trials of the unstable aircraft control laws are scheduled to commence in June 1983, with aircraft centre of gravity being progressively moved aft to introduce negative static stability.

Further minor changes to the control laws are now being defined to optimise the system for flying the aircraft with the leading edge strakes fitted. These trials should take place later in 1983.

6. EXPERIENCE OBTAINED IN DIGITAL FCS DEVELOPMENT AND CERTIFICATION

The principal aim of the Jaguar Demonstrator aircraft programme has been to establish the feasibility of high integrity digital fly-by-wire systems for future production aircraft, and hence reduce the development timescales and risk for such programmes. In fulfilling this aim, comprehensive development, validation and certification activities have been completed to a depth that has confirmed the major problems and identified practical if not optimum solutions.

The novelty of the system is essentially the use of digital computing therefore the principle experience gained has been associated with software design and certification for very high integrity applications. This is summarised in the following paragraphs.

6.1. Software Requirements Definition

Analysis of the 1300 Change Requests raised during the early phases of FRS development shows nearly half were required because of changes to the specification or misinterpretation of the requirements documents. Significant cost and time savings can therefore be achieved by ensuring an accurate and unambiguous definition of requirements early in the programme. Since some changes of definition are inevitable, particularly for a toally new aircraft programme, structuring and segregation of the software to minimise the rework necessitated by the more probable areas of change also improves the efficiency of producing the FRS.

6.2. Software Segregation and Visibility

Visibility of the FRS structure and code is a pre-requisite to subsequent modification potential, analysis of problems found during system testing and subsequent integrity audit of the software. The production of structured, modular software with stringent

procedural, documentation and configuration control can be tedious and is expensive, but no other technique has yet been established which can enable adequate integrity of the resulting software to be determined.

6.3. Programme Store and Run Time Contingency

Minimising programme store and run time constraints reduces the problems of producing the first issue of a real time software programme. Even greater benefits are found when modifications are subsequently required. Therefore to keep total development costs to an acceptable level, and also maintain visibility of the final software, considerable attention must be given to hardware capability and software design and structure. Cost effective contingency allowances must be made available within the segregated programme store and the segmented software run time structure to allow future modification without the knock on effects of restructuring hardware and/or software or total re-allocation of the programme within the store devices.

6.4. Integrity Audit

The Jaguar Fly-By-Wire programme has developed integrity audit techniques and procedures which have enabled the aircraft to be cleared for flight without having to compromise any of the original requirements. The success of this aspect of the programme has been dependent on many factors including:-

- Independent auditors
- Structuring the integrity analysis to assume perfect implementation, then assessing the probability of defects in the identified critical implementation features.
- Correlation of results from both 'top down' and 'bottom up' analyses
- Constructive use of emulation and control flow analysis techniques
- Dedicating Senior engineering resources to complete a thorough integrity appraisal.

6.5. Development Tools

The task of developing and validating high integrity digital systems can only be achieved in practical timescales if adequate tools are made available. Powerful, efficient and well proven software tools are necessary to contain the task of software production, testing and configuration control. Sophisticated rig facilities are essential to enable thorough testing of the full system executing representative flight tasks in real time. Reliable hardware, with dependable BIT, supported by comprehensive data acquisition and processing facilities enable extensive testing to be carried out in realistic timescales. The hardware and software techniques developed by MAv, complemented by the BAe developed rigs, emulation and data acquisition systems, have identified and assembled a powerful capability for developing future systems.

7. DEVELOPMENTS FOR THE FUTURE

Plans are now being considered for extending the role of the Jaguar Demonstrator aircraft beyond the strakes flight test programme. However any resultant programme is likely to use the aircraft to investigate control techniques rather than concentrate on FCS development. In general, therefore, further software development is expected to be cost constrained to minimum changes within the existing definition, structure and production techniques.

Extensions, adaptations and enhancements of these techniques are therefore being associated with new programmes such as P110/ACA. Building upon the experience established prior to, and during, the Jaguar programme, the following software specification, organisation and coding concepts are now being evaluated.

7.1. Software Requirements Definition

The software requirements definition can introduce problems in three ways - errors, omissions and ambiguities. Improving the methods of definition can do nothing to prevent errors resulting from incorrect assessment of the aircraft characteristics or the control task, but it should be possible to reduce the remaining sources of problems. Most of these are introduced at the boundaries between data bases. Transfer of information from the control law designer, to the requirements documentation, thence to the detail software control specification and eventually the code and test processes, all potentially introduce translation errors, misinterpretations and omissions. Consideration has, therefore, been given to techniques which improve the visibility of these translation processes and provide scope for more automated correlation between the initial requirements and the final code. Writing the initial requirements document in machine executable statements enables the definition to be exercised against the aircraft model, and subsequently the performance of the final code can be checked against the same model. Correlation of the results should then rapidly detect any errors that have been introduced. Adoption of a more 'top down' approach to producing software requirements documents should minimise omissions within the definition and should also provide a more ordered and perhaps more efficient structure.

7.2. Segregation

Extension of the software partitioning already practised can provide further benefits, particularly where the FRS development is to be carried out by more than one organisation e.g. task sharing between avionics supplier and airframe company. As a next step, segregation of the software into two or three essentially autonomous sections is proposed. These would cover for example Executive and Data I/O (type A), Data consolidation and system monitoring (type B) and Control Law tasks (type C). Each would be allocated segments of the programme store and frame run time, with communication via nominated locations within the scratchpad. All work space locations would be read/write protected to minimise illegal data transfer in the event of hardware faults or software design errors. With this structure the software can be developed by separate teams with reduced short term interaction. Since the type A, and to a slightly lesser extent the type B, software will change very little for a given system, the control law changes can be contained within the type C software (perhaps 30% of the programme) with very high confidence that the integrity of the remainder of the programme has not been compromised.

7.3. Task Orientated Programme Language

The standard macro library used for the Jaguar FBW software is being extended to cover the majority of the tasks required by the control law designer. By using macro names and parameters which are familiar to the control law designer, incorporating scaling functions, and providing data fetch and store facilities a task orientated Flight Control Language (FLICOL) has been created. Figure 14 shows an example of a control law path written in this language demonstrating the visibility that can be achieved.

```
     Kr :=0.2
     KXI :=0.6
     KDDay:=0.05
     GET r
     GAIN (Kr)
F10  WASHOUT (3.33)
     SCALE (5.0)
     PUT HOLD 1
     GET XI
     GAIN (KXI)
     SUM (HOLD 1)
     PUT HOLD 2
     GET ay
     GAIN (KDDay)
F11  WASHOUT (20.0)
     SCALE (2.0)
     SUM (HOLD 2)
F12  LAG 1 (0.04)
F13  SCHED1 (150.0, 400.0, 1.0, 0.1, 650.0, V)
     PUT ZETA
```

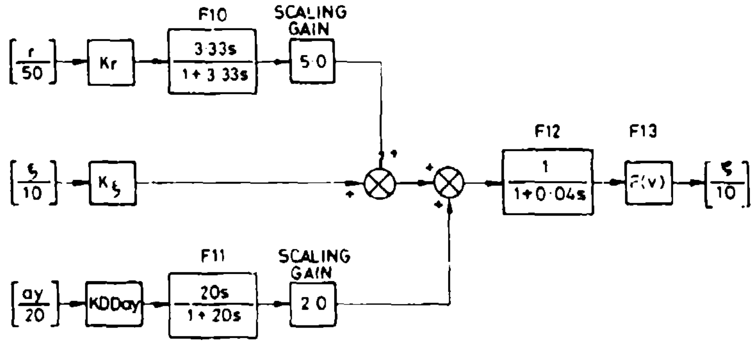

Figure 14 Example of Flight Control Language (FLICOL) Statements

The support tools for this language are based on those presently developed and well proven, providing a relatively simple translation to the selected instruction set of the target processor. These tools can include engineering calculations to relieve the programmer of tasks associated with defining filters, voter monitors, rate limits etc. which are functions of iteration rates.

FLICOL can also be developed as a systems simulation language. This could lead to a situation where the control laws developed on the simulator can be directly translated to the programme for the target processor without the need for source code changes and thus reduces the possibility of introducing errors or misinterpretations.

7.4. High Order Languages

The macro assembler language is considered a highly visible, efficient and safe approach to producing high integrity software, particularly for special purpose processors with instruction sets optimised for flight control applications.

The use of general purpose microprocessors, high order languages and compilers for high integrity applications has caused concern because of the lack of visibility of the device structure, microprogram and compiler 'optimisation' routines. With the development of task orientated microprocessors such as those implementing MIL-STD-1750A, and corresponding languages with more formal verification such as JOVIAL and perhaps ADA these limitations are being minimised. Future use of these, in applications where standardisation of hardware and software production methods are very significant, is being pursued.

ACKNOWLEDGEMENTS

The Jaguar Fly-by-Wire programme has been carried out with the support of the Procurement Executive Ministry of Defence.

The author acknowledges the special team effort that this programme has involved. Success depends upon the combined efforts of British Aerospace, Dowty Boulton Paul and Marconi Avionics. In particular the programme would not have commenced or been able to continue without the enthusiastic support of the British Ministry of Defence and RAE Farnborough. His thanks are extended to colleagues in British Aerospace and the Combat Aircraft Controls Division of Marconi Avionics for assistance provided in the preparation of this paper.

REFERENCES

1. R.E.W. Marshall, K.S. Snelling, J.M. Corney, 'The Jaguar Fly-[illegible]-Wire Dem[illegible]tor Integrated Flight Control System', 1981 Proceedings of Advanced Flig[illegible]ols Symposium, USAF Academy.
2. E. Daley, R.B. Smith, 'Flight Clearance of the Jaguar Fly-By-Wire Airc[illegible]982 Proceedings of the Royal Aeronautical Society Avionics Systems Group [illegible]posium 'Certification of Avionic Systems'.
3. T.D. Smith, C.J. Yeo, R.E.W. Marshall, 'Ground and Flight Testing on the Fly-By-Wire Jaguar Equipped with a Full Time Quadruplex Digital Integrated Flight Control System'. AGARD 35th Guidance and Control Panel Symposium, Portugal 82, 'Advances in Guidance and Control Systems'.
4. P.A. Doggett, I. P. MacDiarmid (BAe Warton), 'Lightning Protection Design and Lightning Threat Flight Clearance of a Fly-by-Wire Flight Control System for an Unstable Aircraft' (to be published).
5. R.H. Evans, J. Bishop, (Royal Aircraft Establishment), 'Induced Transients in Simulated Lightning Test of the Fly-by-Wire Jaguar Aircraft' (to be published).